Compsci 6/101: PFTW

- APTs, Assignments, Tools
  - What, How, When, Why
  - How to get help and when to get it
- Review types and operations on types
  - Names and types: int, float, long, *string*, file, list,…
  - Operations: *, +, -, /, %, ** [:]
- Functions: organizing code (and ideas)
  - Functions, modules, indentation, naming
  - Parameterization to generalize solutions
  - Return values to caller of function

Functions: abstractions over code

- Naming something gives you power
  - How do you find absolute value?
  - What is length of a string?
- We can write and call functions
  - Re-use and/or modify
  - Store in module, import and re-use functions
  - Import standard modules and use functions from them
- Functions can (should?) return a value
  - We've seen len return an int, what about file.read()?
  - Other functions return Strings, floats, or other types

Re-use: Counting words in file

```
def wordCount(filename):
    file = open(filename)
    str = file.read()
    words = str.split()
    return len(words)

name = "/data/romeo.txt"
print "# words in",name,
print "=",wordCount(filename)
```


Anatomy of a Python function

```
def name(params):
    body
```

- Define a function, provide a name, provide parameters, provide a function body
  - How to decide on name?
  - Do we need parameters?
  - What does body of function do
- Functions provide a named abstraction over code
  - Huh? What is `math.isinf(..)` or `math.sqrt(..)`?

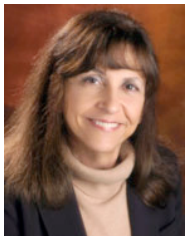
Revisiting functions

- Python Heron's formula or BMI (Body Mass Index)
  - What's the name of the function
  - What are parameters that enable function use/call
- How do we write and test APTs
  - What's the name of the function
  - What are parameters that enable function use/call
  - Who writes the function? Who calls the function?
- How will you decide on these things in writing your own code?

Design, Implementation, Testing

- Designing Python code and functions
  - What do you want the code to do?
  - Other aspects of code, e.g., portability, efficiency, size, …
  - Understand how to solve a problem without computer
- Implementing design in code
  - Translation of ideas into vocabulary of Python
  - We don't have a large vocabulary, but it will grow!
- Testing code, functions
  - How do you know when function is right?
  - What confidence can testing provide?
  - APT testing is similar to Unit Testing (well known)

Nancy Leveson: Software Safety



- Mathematical and engineering aspects, invented the discipline
  - Air traffic control
  - Microsoft word

  "There will always be another software bug; never trust human life solely on software" *huffington post?*
- Therac 25: Radiation machine
  - http://en.wikipedia.org/wiki/Therac-25
  - http://bit.ly/5qOjoH
- Software and steam engines

String functions in Python

- What is a string?
  - *Immutable* sequence of 'characters' with operations on sequence
- Strings are objects, some functions 'builtin'
  - Methods: `strip`, `startswith`, `find`, `count`, …
  - Functions on strings: `len(.)`, return strings: `f.read()`
  - Strings are immutable: `"hello".replace("h","j")`
- Slicing operator for strings
  - What is an index, where does indexing begin?
  - Value of `s[a:b]` compare `s[a:]` compare `s[:b]`

Language and Problems in Context

- Convert Romeo and Juliet to Piglatin
  - What else could we do with Shakespeare's plays?
  - How do we convert HTML to text?
  - Convert English to Russian, Italian, Spanish, …
  - Remove all $$ from salary data in a file
- How do you determine if 2040 is a leap year?
  - Any year is a leap year?
- How do we make an image larger, more red, …
  - What is an image? How do read it? Convert it? Access it?

What years are leap years?

- 2000, 2004, 2008, …
  - But not 1900, not 2100, yes 2400!
  - Yes if divisible by 4, but not if divisible by 100 unless divisible by 400! (what?)

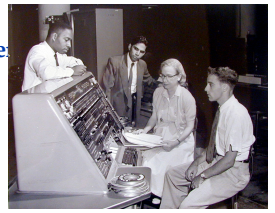
```
def is_leap_year(year):
    if year % 400 == 0:
        return True
    if year % 100 == 0:
        return False
    if year % 4 == 0:
        return True
    return False
```

- There is more than one way to skin a cat, but we need at least one way

Python if statements and Booleans

- In python we have if: else: elif:
  - Used to guard or select block of code
  - If guard is True then, else other
- What type of expression used in if/elif tests?
  - ==, <=, <, >, >=, !=, and, or, not, in
  - Value of expression must be either True or False
  - Type == bool, George Boole, Boolean,
- Examples with if
  - String starts with vowel
  - Rock, paper, scissors (!aka Rochambeau) winner

Grace Murray Hopper (1906-1992)



- "third programmer on world's first large-scale digital computer"
  - US Navy: Admiral

"It's better to show that something can be done and apologize for not asking permission, than to try to persuade the powers that be at the beginning"

- ACM Hopper award given for contributions before 35

  2004: Jennifer Rexford

  2009: Tim Roughgarden